REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
09-01-2015	Final Report	1-May-2009 - 31-Oct-2014

4. TITLE AND SUBTITLE
Final Report: Rotman Lens Sidewall Design and Optimization with Hybrid Hardware/software Based Programming

5a. CONTRACT NUMBER
W911NF-09-1-0123

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Ozlem Kilic

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
The Catholic University of America
620 Michigan Ave, NE

Washington, DC 20064 -0001

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
51920-EL.17

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This final report provides a summary of the tasks completed under this contract:

1. Hybrid bio-inspired optimization: This task involves developing a hybrid bio-inspired optimization algorithm which enhances the Artificial Immune System (AIS) algorithm with techniques borrowed from PSO and GA. Preliminary results with three challenging mathematical test functions were shown to be promising in terms of the performance of the enhanced AIS (EAIS) algorithm.

15. SUBJECT TERMS
hybrid numerical electromagnetics, MoM, FMM, CPU-GPU, Rotman lens

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Ozlem Kilic
UU	UU	UU	UU		19b. TELEPHONE NUMBER 202-319-5261

Report Title

Final Report: Rotman Lens Sidewall Design and Optimization with Hybrid Hardware/software Based Programming

ABSTRACT

This final report provides a summary of the tasks completed under this contract:

1. Hybrid bio-inspired optimization: This task involves developing a hybrid bio-inspired optimization algorithm which enhances the Artificial Immune System (AIS) algorithm with techniques borrowed from PSO and GA. Preliminary results with three challenging mathematical test functions were shown to be promising in terms of the performance of the enhanced AIS (EAIS) algorithm.
2. Hybrid method for electrically large structures: This task involves developing a hybrid code that can combine analytical, numerical and asymptotic techniques as they best fit to the problem at hand to simulate electrically large structures.
3. Rotman lens applications: This task involves the application of hybrid techniques to model printed Rotman lenses.
4. Parallelization of full-wave techniques on hybrid platforms: This task involves developing a hybrid MoM-FMM code that utilizes CPU as well as GPU to handle problem sizes in the order of 10 Million unknowns.

By utilizing hybrid techniques in addition to hybrid programming platforms, we have been able to demonstrate that we can solve electromagnetic problems with 10 Millions of unknowns with the 13 node GPGPU cluster we have in place.
OUr experimental results validate the scalability of our code, achieving a close to linear performance as the number of nodes are increased.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper
10/07/2014 14.00	Vinh Dang, Quang Nguyen, Ozlem Kilic. GPU Cluster Implementation of FMM-FFT for Large-Scale Electromagnetic Problems, IEEE AWPL , (06 2014): 1259. doi:
10/07/2014 13.00	Vinh Dang, Quang Nguyen, Ozlem Kilic. Fast Multipole Method for Large-Scale Electromagnetic Scattering Problems on GPU Cluster and FPGA-Accelerated Platforms, ACES Journal, SPecial Issue, (12 2013): 1187. doi:
TOTAL:	**2**

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper
TOTAL:	

Number of Papers published in non peer-reviewed journals:

(c) Presentations

Number of Presentations: 0.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper	
10/18/2012	3.00	Quang Nguyen, Ozlem Kilic. Enhanced Artificial Immune System Algorithm Using PSO and GA Principles, ACES 2012. 10-APR-12, . : ,
TOTAL:	**1**	

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
09/11/2013	9.00	Ozlem Kilic, Esam El-Araby, Vinh Dang,, uang Nguyen. Single Level Fast Multipole Method on GPU Cluster for Electromagnetic Problems, URSI USNC. 03-JAN-13, . : ,
09/11/2013	10.00	Quang Nguyen, Ozlem Kilic, Esam El-Araby, Vinh Dang. Fast Multipole Method for Large-Scale Electromagnetic Scattering Problems using High Performance Computers, ACES 2013. 24-MAR-13, . : ,
09/11/2013	11.00	Quang Nguyen, Vinh Dang, Ozlem Kilic. Graphics Processing Unit Accelerated Fast Multipole Method - Fast Fourier Transform, IEEE AP-S. 07-JUL-13, . : ,
10/07/2014	15.00	Quang Nguyen, Ozlem Kilic. Electromagnetic Scattering from Multiple Domains Using a Hybrid Numerical and Analytical Solution, ACES 2014. , . : ,
10/07/2014	16.00	Quang Nguyen, Ozlem Kilic. Electromagnetic Scattering from Multiple Domains Using a Hybrid Numerical and Analytical Solution, ACES 2014. , . : ,
10/18/2012	4.00	Ozlem Kilic, Esam El-Araby, Vinh Dang. Investigating Interferometric Imaging in Random Media using CUDA and Jacket Environments for GPUs, ACES 2012. 10-APR-12, . : ,
TOTAL:	**6**	

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
09/11/2013	12.00	Vinh Dang, Ozlem Kilic, Esam El-Araby, Quang Nguyen. Parallelizing Fast Multipole Method for Large-Scale Electromagnetic Problems using GPU Clusters, IEEE AWPL (06 2013)
10/18/2012	2.00	Esam El-Araby, Ozlem Kilic, Vinh Dang. Exploiting FPGAs and GPUs for Electromagnetics Applications: Interferometric Imaging in Random Media Case Study, ACES Journal, SPecial Issue (02 2012)
10/18/2012	1.00	Ozlem Kilic, Esam El-Araby, Quang Nguyen, Vinh Dang. Bio-Inspired Optimization for Electromagnetic Structure Design Using Full-Wave Techniques on GPUs, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (08 2012)
TOTAL:		**3**

Number of Manuscripts:

Books

Received	Book
TOTAL:	

Received	Book Chapter
TOTAL:	

Patents Submitted

Patents Awarded

Awards

Graduate Students

NAME	PERCENT_SUPPORTED	Discipline
Nghia Tran	1.00	
FTE Equivalent:	**1.00**	
Total Number:	**1**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Ozlem Kilic	0.08	
FTE Equivalent:	**0.08**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period:	0.00
The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:.....	0.00
Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):.....	0.00
Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:.....	0.00
The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense	0.00
The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:	0.00

Names of Personnel receiving masters degrees

NAME

Total Number:

Names of personnel receiving PHDs

NAME

Total Number:

Names of other research staff

NAME PERCENT_SUPPORTED

FTE Equivalent:

Total Number:

Sub Contractors (DD882)

Inventions (DD882)

Scientific Progress

The following tasks were completed as part of the program:

1. Hybrid bio-inspired optimization: This task involves developing a hybrid bio-inspired optimization algorithm which enhances the Artificial Immune System (AIS) algorithm with techniques borrowed from PSO and GA. Preliminary results with three challenging mathematical test functions were shown to be promising in terms of the performance of the enhanced AIS (EAIS) algorithm.
2. Hybrid method for electrically large structures: This task involves developing a hybrid code that can combine analytical, numerical and asymptotic techniques as they best fit to the problem at hand to simulate electrically large structures.
3. Rotman lens applications: This task involves the application of hybrid techniques to model printed Rotman lenses.
4. Parallelization of full-wave techniques on hybrid platforms: This task involves developing a hybrid MoM-FMM code that utilizes CPU as well as GPU to handle problem sizes in the order of 10 Million unknowns.

Pleases see the attachment for a detailed description on each task.

Technology Transfer

FINAL REPORT

December 2014

ARO Grant W911 NF-09-1-0123, Modification P00004

Rotman Lens Sidewall Design and Optimization with Hybrid Hardware/Software Based Programming

The following tasks were completed as part of the program:

1. **Hybrid bio-inspired optimization**: This task involves developing a hybrid bio-inspired optimization algorithm which enhances the Artificial Immune System (AIS) algorithm with techniques borrowed from PSO and GA. Preliminary results with three challenging mathematical test functions were shown to be promising in terms of the performance of the enhanced AIS (EAIS) algorithm.
2. **Hybrid method for electrically large structures**: This task involves developing a hybrid code that can combine analytical, numerical and asymptotic techniques as they best fit to the problem at hand to simulate electrically large structures.
3. **Rotman lens applications**: This task involves the application of hybrid techniques to model printed Rotman lenses.
4. **Parallelization of full-wave techniques on hybrid platforms**: This task involves developing a hybrid MoM-FMM code that utilizes CPU as well as GPU to handle problem sizes in the order of 10 Million unknowns.

Task 1: Hybrid bio-inspired optimization

1A. Enhanced Artificial Immune System Algorithm

Bio-inspired optimization techniques rely on agents that independently sample the optimization space until a desired solution or the maximum number of iterations is reached. The initial step is purely random, hence requires no a-priori guess of the solution. More intelligence is added to the search as time progresses based on accumulated knowledge on the search domain. This intelligence is based on the computation of a cost function, which is a measure of how well each agent has performed with respect to the desired solution; with high costs referring to poor solutions. The general principles of bio-inspired optimization are shown in Fig. 1.

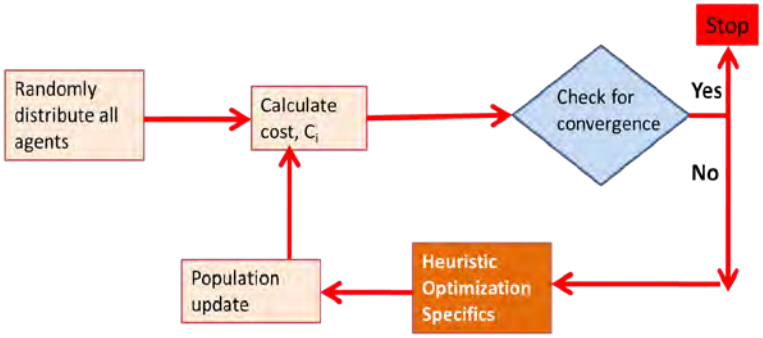


Fig. 1. A general block diagram of bio-inspired optimization methods.

The computation of the cost function is the most numerically intensive part, and those algorithms which achieve a good solution with fewer number of cost computations are deemed more efficient. Employing more agents helps reduce the number of iterations to converge, but naturally requires a higher number of cost computations per iteration. Therefore, the total number of cost computations required to reach a solution is an objective means of testing the effectiveness of these algorithms.

The AIS optimization is based on the clonal selection principles of our immune response to potential disease generating metabolisms, and simulates our body's defense system against viruses. Our adaptive immune system produces antibodies whose purpose is to bind to any antigen that it recognizes. For engineering applications, antibodies represent a possible solution to the optimization problem. The optimization space is discretized in order to emulate the binary form of gene behavior. The heuristic optimization specifics step in Fig. 1 is replaced by four steps in AIS: cloning, mutation, combination and sorting as shown in Fig. 2.

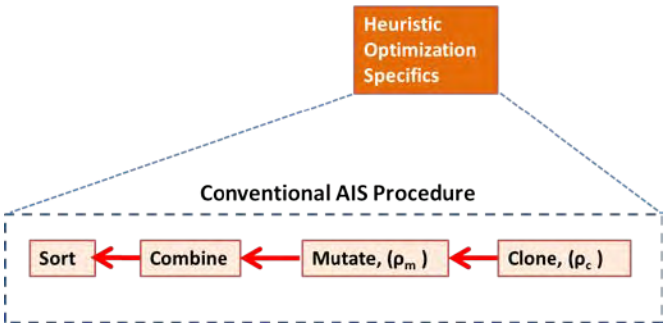


Fig. 2. Conventional AIS procedure steps.

The AIS algorithm begins the search with an initial set of random guesses of N_a antibodies; i.e. potential solutions. Each solution is represented by a bit string with a length equal to the product of the number of optimization variables and the number of bits per variable, N_b. The antibodies are sorted based on their cost values; with lower cost solutions ranking higher in the list. Then an

intelligent random search begins for the next set of solutions through the cloning and mutation processes, which are carried out at different rates defined by coefficients ρ_c, and ρ_m respectively. Based on ρ_c, a duplicate set of N_{ck} antibodies is created for the k^{th} antibody, such that good solutions are cloned more than poor ones. For the mutation step, a number of bits in the string are flipped randomly. The number of bits to be flipped is directly proportional to the index number k of each antibody in the list, and is calculated based on the mutation rate. Finally, the cloned and mutated sets are combined with the original set and sorted again. The basic steps of the AIS algorithm are demonstrated in Fig. 3. The top N_a antibodies of the combined set are then selected for the next iteration. The process continues until either a desired solution or the maximum number of iterations is reached. Further details of AIS can be found in [1].

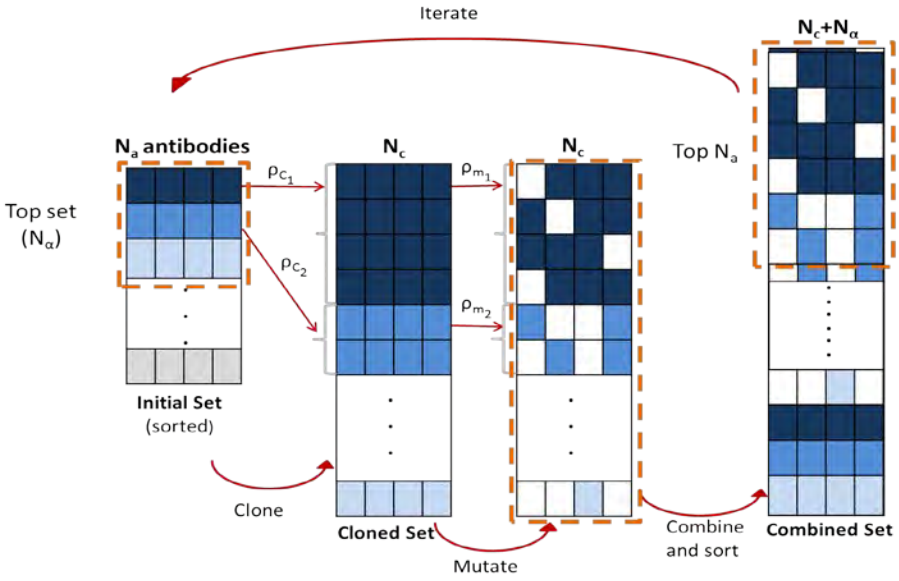


Fig. 3. Conventional AIS antibody production.

In challenging problems, AIS can stagnate, and all good solutions may differ by only a few bits. By incorporating modifications to the conventional AIS algorithm, which bring more intelligence to the mutation stage, as well as by introducing concepts from other algorithms, its performance can be enhanced. We refer to the revised form of AIS as the enhanced AIS (EAIS) in the rest of the report.

EAIS specific procedures are shown in Fig. 4, where brown and blue boxes indicate a modification to the existing process, and the newly introduced step, respectively. Mutation is no longer carried out randomly, but is inspired by the concept of velocity towards the global best as in PSO [2]. Furthermore, a cross-over step is added based on the concept of producing children's genes in GA, [3].

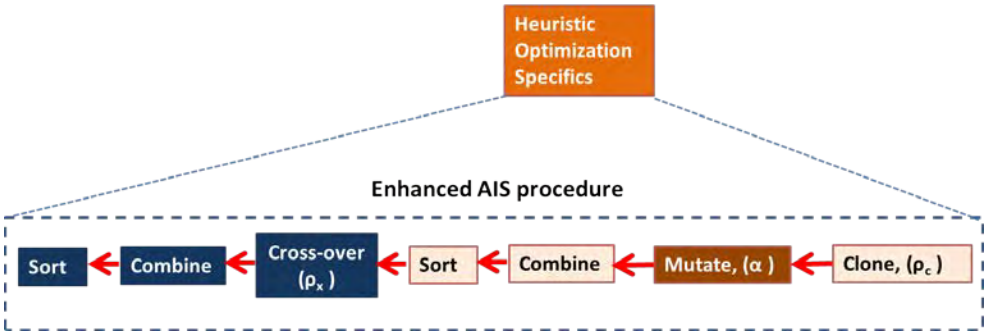


Fig. 4. Enhanced AIS procedure steps.

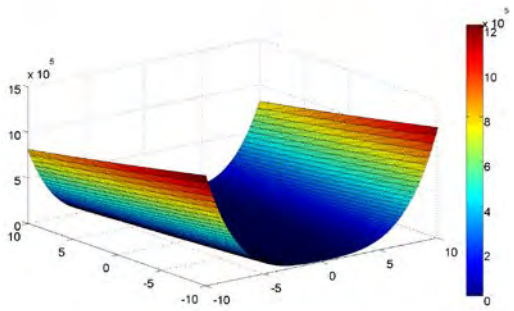
1B. Performance Study for EAIS

We investigated how EAIS compares to other bio-inspired optimization algorithms (e.g. PSO, AIS and ACO) by applying them to three mathematical test functions; namely Rosenbrock, Rastrigin, and Griewank. Each of these functions presents a unique set of challenges, and can be used to test the robustness of these algorithms. They are multi-dimensional functions, with a minimum value, which is zero, at $\bar{x} = 0$ for Griewank and Rastrigin functions, and $\bar{x} = 1$ for the Rosenbrock function. The expressions of these functions are given below, and their behaviors are demonstrated in Fig. 5 for the two dimensional case, i.e. N = 2.

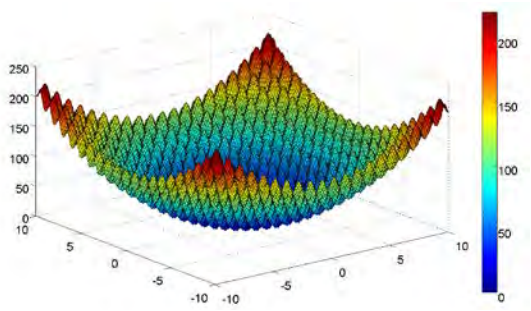
$$f_{Rose}(x) = \sum_{i=1}^{N} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), -10 \le x_i \le 10.$$

$$f_{Rast}(x) = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i) + 10), -10 \le x_i \le 10.$$

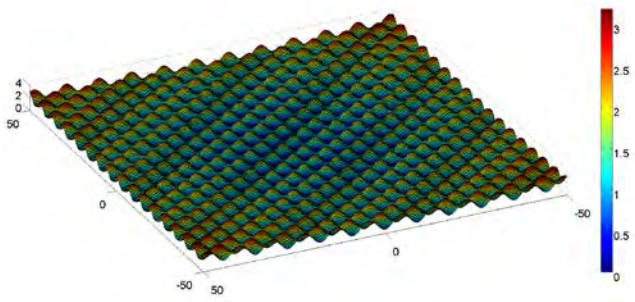
$$f_{Grie}(x) = \frac{1}{4000} \sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1, -50 \le x_i \le 50.$$



(a) Rosenbrock



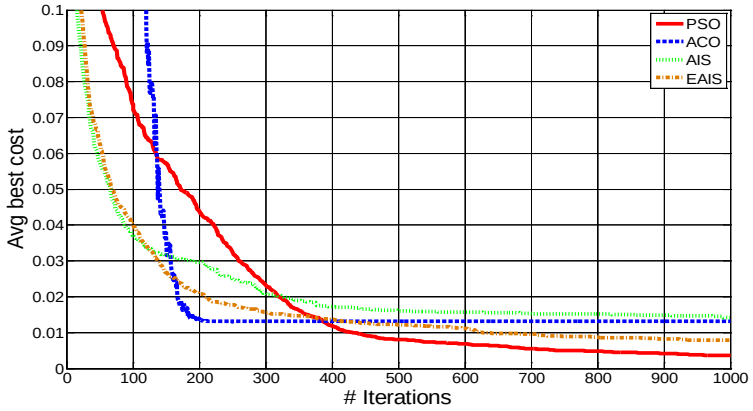
(b) Rastrigin



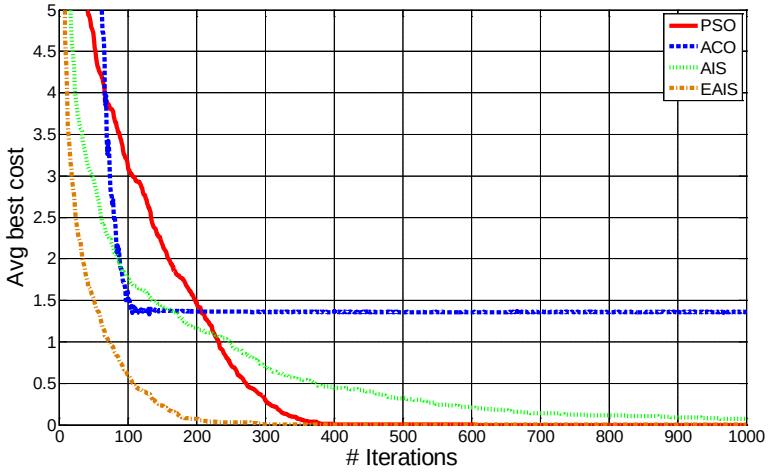
(c) Griewank

Fig. 5. Mathematical test functions in their 2-D forms.

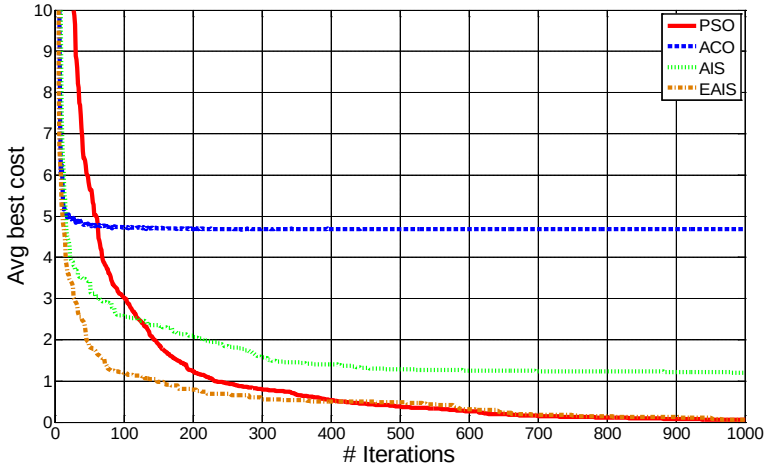
To investigate the performance of the algorithms, we observe how each of them converges to the minimum value as time progresses. A plot of the average best cost for 100 simulations as a function of the number of iterations is shown in Fig. 6 for each test function. We observe that EAIS is the fastest to converge to a low cost for all of the test functions, with PSO being the second contender.



(a) Griewank function



(b) Rastrigin function



(c) Rosenbrock function

Fig. 6. Performance comparison of the four optimization algorithms.

1C. Parallel Implementation of EAIS on GPU system.

Calculation of the cost function, which is the most time-consuming part of the algorithm, need to be done for each agent. Since each agent is independent, it can be run in parallel. Each GPU can handle a certain number of agents based on the number of GPUs available in the system, the memory of each GPU as well as the complexity of the cost function. Fig. 7 shows the flowchart of the EAIS implemented with *N* GPUs.

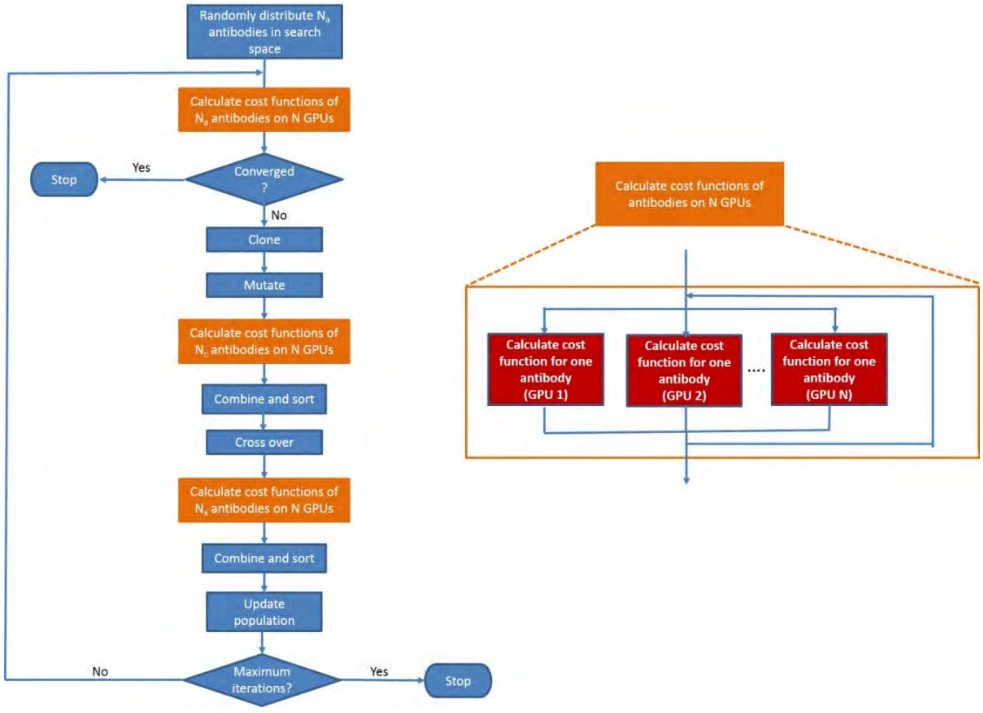


Fig. 7. EAIS flowchart with *N* GPUs.

Task 2: Hybrid method for electrically large structures

2A. Hybrid Method using Numerical and Analytical Solution

Hybrid methods often combine at least two numerical techniques to efficiently simulate electrically large structures without a significant loss of accuracy, [4]-[5]. In this report, we develop a new hybrid method using a combination of numerical and analytical solutions. It is based on dividing the original object into smaller sub-domains. The sub-domains, which may contain arbitrarily shaped objects, can be solved by numerical techniques such as MoM. If a sub-domain contains canonical objects, analytical techniques (such as the Mie solution for spherical objects) can be used. Asymptotic techniques can also be used. Each sub-domain is solved independently, and then the interactions between the sub-domains are accounted for iteratively. We aim to utilize the unique features of each sub-domain to reduce the computational complexity, while still preserving the accuracy. A general block diagram of this hybrid method for the case of two sub-domains is shown in Fig. 8.

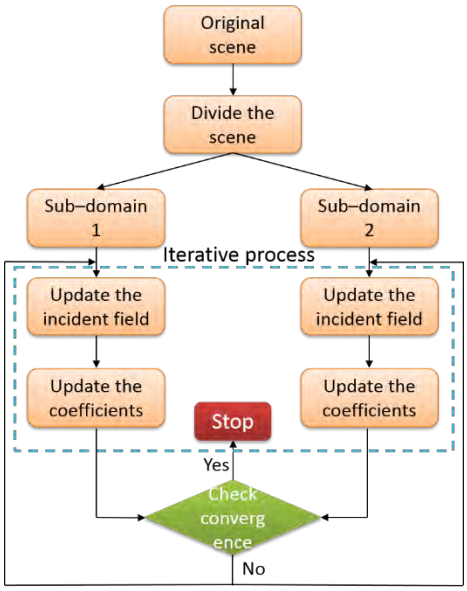
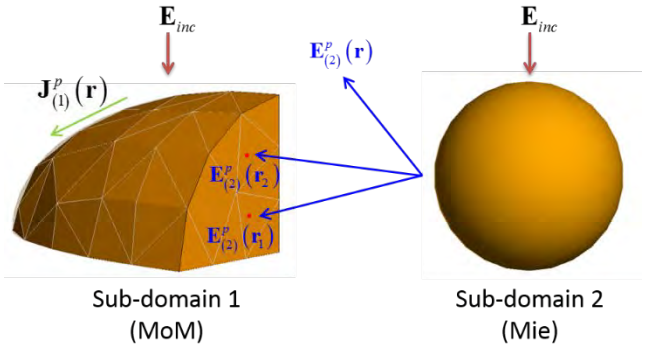


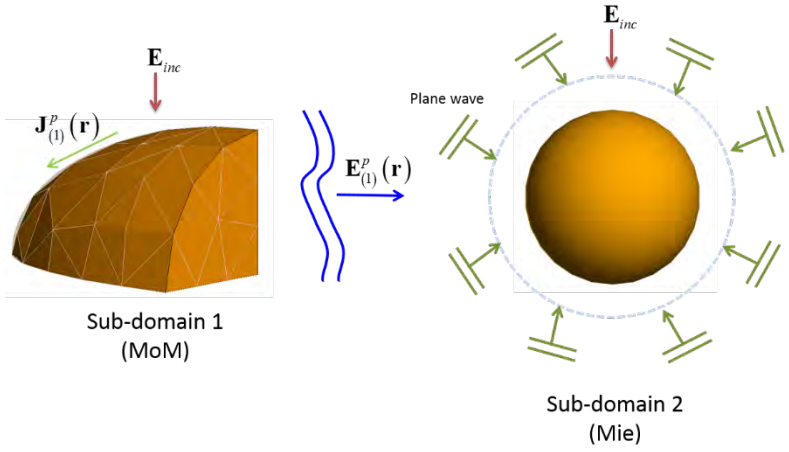
Fig. 8. A general block diagram of the hybrid method.

The details of the iterative scheme are discussed for the case of two sub-domains below in Fig. 9. At the first iteration, a plane wave, $\mathbf{E}_{inc}$, illuminates the computational domain, and is assumed to be the only source of excitation. At iteration *p*, the total incident field for sub-domain 1 (as shown in Fig. 9a) is equal to the sum of original incident field, $\mathbf{E}_{inc}$, and the radiated fields from sub-domain 2, $\mathbf{E}^{p}_{(2)}(\mathbf{r})$. Similarly, the total incident field for sub-domain 2 is equal to the sum of

original incident field, $\mathbf{E}_{inc}$, and the radiated fields from sub-domain 1, $\mathbf{E}^{p}_{(1)}(\mathbf{r})$. However, the solution for sub-domain 2 (Mie scattering) requires the incident field to be in the form of plane waves. Therefore the radiated fields from sub-domain 1, $\mathbf{E}^{p}_{(1)}(\mathbf{r})$, must be decomposed into local plane waves illuminating sub-domain 2, as shown in Fig. 9b. This is necessary when any sub-domain utilizes a method based on plane wave illumination. Further details for the hybrid techniques can be found in [6]-[7].



(a) From sub-domain 2 to sub-domain 1



(b) From sub-domain 1 to sub-domain 2

Fig. 9. Interfacing between sub-domains.

2B. Numerical Results

In this section, we provide a numerical example to validate the accuracy of the hybrid method (MoM-Mie in this case). We compare our results with the full-wave solution using the commercial software package FEKO. Fig. 10 shows the configuration for the test case, where two small conducting spheres with diameters of 1.5λ are separated by a distance $d = 2.5\lambda$ from

a larger conducting sphere with diameter of 3λ . The spheres are placed along the *x*-axis, and are excited by a *z*-polarized field plane wave along the $\vec{x}$ direction.

The scattering problem is decomposed into two sub-domains. Sub-domain 1 consists of the two smaller spheres and sub-domain 2 consists of the large sphere. For sub-domain 1, we employ MoM while for sub-domain 2, we consider two different cases: MoM and Mie scattering. Once the fields are generated, the mutual interactions between the spheres take place by applying the iterative procedure in the same manner described in Section A. The incident field on sub-domain 2 is decomposed into N = 961 plane waves. We show the backscattered radar cross section (RCS) results for the problem in Fig. 11 for the three cases. The dotted line shows the results for the entire problem using FEKO. The hybrid method results for the two cases are shown with the dashed line for the MoM-Mie combination, and with the solid line for the MoM-MoM combination. We observe that both hybrid methods offer a good agreement with FEKO. Both calculations converge after 9 iterations.

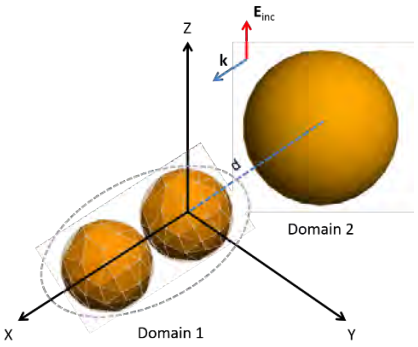


Fig. 10. Geometry of the test case with three conducting spheres.

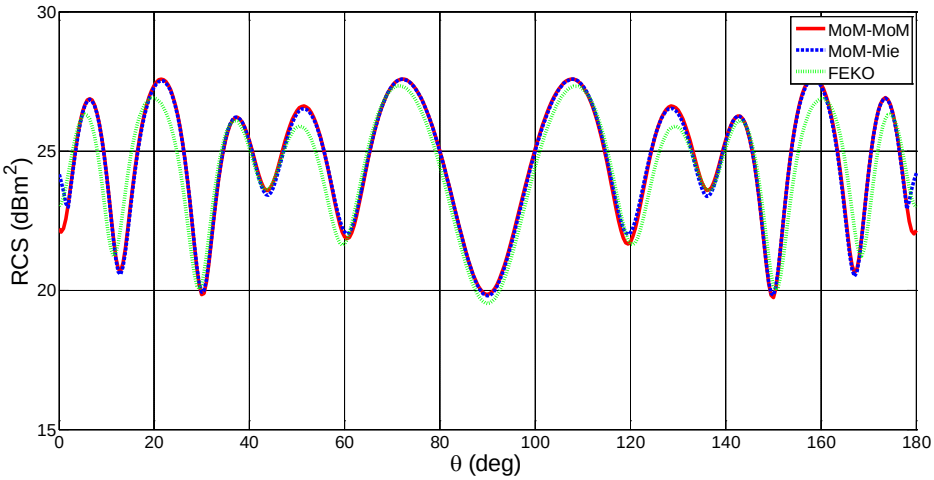


Fig. 11. Calculated RCS for the test case.

Task 3: Rotman lens applications

3A. Analysis Approach for Rotman Lens

The classical methods for analysis and design of Rotman lenses, [8]-[9] or any different microwave lens are based on geometrical optics. However, the mutual coupling between ports, the multiple reflections within the lens cavity and the discontinuity at the junctions between the lens and the transmission lines at the beam or radiating ports are not incorporated. To overcome these limitations, we use an alternative two dimensional field analysis approach, which is based on the contour integral method [10] to analyze the lens. In this method, first the impedance matrix, Z, of the lens is calculated. Then the S-parameters, which yield the reflection, coupling, and transmission coefficients between various beam and radiating ports, are obtained by transforming the Z matrix. Finally the radiation characteristics of the lens are computed. The details of the method are discussed below.

We consider an arbitrarily shaped planar microstrip configuration as shown in Fig. 12a. In order to obtain the RF voltage along the periphery, the wave equation is converted to into the contour integral form. The potential at an arbitrary point on the periphery satisfies the following integral equation (1) which is derived from Weber's solution for cylindrical waves [10].

$$V(s) = \frac{1}{2j}\oint_C \left[k\cos(\theta) H_1^{(2)}(kr) V(s_0) - j\omega\mu dH_0^{(2)}(kr) i_n(s_0) \right] ds_0 \qquad (1)$$

where $H_0^{(2)}$ and $H_1^{(2)}$ are the zero order and the first order of the second kind of the Hankel functions, respectively. The variable r denotes the distance between points M and L, θ denotes the angle between the vector from point M to L and the normal vector at point L, and i_n is the current density along the periphery. Equation (1) gives the relation between the RF voltage and the RF current along the periphery.

To solve equation (1) numerically, the periphery is divided into N segments numbered as $1, 2, \ldots, N$ corresponding to segment widths of $W_1, W_2, \ldots, W_N$, respectively as shown in Fig. 12b. By assuming the electric and magnetic field intensities are constant over the width of each segment, the integral equation in (1) can be rewritten into a system of matrix equations.

$$V = U^{-1}HI \qquad (2)$$

The matrix U and H can be found in [10]. From the above relations, the impedance matrix of the equivalent periphery is obtained as

$$Z = U^{-1}H \qquad (3)$$

(a) (b)

Fig. 12. Arbitrary planar configuration analyzed by the contour integral approach.

3B. Segmentation using *Z* Matrix

The contour integral method discussed in the previous section can be applied for any arbitrary two dimensional planar structure. However, in many practical applications, the two dimensional planar structure can be decomposed into several segments which themselves have simpler shapes such as squares, rectangles, triangles, circular sectors and so on. For these shapes, employing an analytical approach provides better efficiency compared to contour integral approach. In this section the segmentation method that combines the *Z* matrix of each segment to give the entire Z matrix is discussed.

In the segmentation method, the overall *Z* matrix can be decomposed as

$$\begin{bmatrix} V_p \\ V_c \end{bmatrix} = \begin{bmatrix} Z_{pp} & Z_{pc} \\ Z_{cp} & Z_{cc} \end{bmatrix} \begin{bmatrix} I_p \\ I_c \end{bmatrix} \tag{4}$$

where *p* and *c* are the externally and internally connected ports, respectively as shown in Fig. 13. *V*, *Z* and *I* are the voltage, current and impedance matrices at the corresponding ports. The interconnection constrains are as follows: the voltages at two connected ports are equal, and the sum of currents at the two connected ports is zero. These conditions can be formulated as

$$\begin{aligned} \Gamma_1 V_c &= 0 \\ \Gamma_2 I_c &= 0 \end{aligned} \tag{5}$$

where Γ_1 and Γ_2 are two matrices with $c/2$ rows and c columns describing the connections. Γ_1 and Γ_2 can be expressed as

$$\begin{aligned} \Gamma_1 &= [I \,|\, -I] \\ \Gamma_2 &= [I \,|\, I] \end{aligned} \tag{6}$$

where *I* is identity matrix. Combining (5) and (6), the overall impedance Z matrix can be derived as

$$Z_p = Z_{pp} - Z_{pc}\begin{bmatrix}\Gamma_1 Z_{cc}\\ j\Gamma_2\end{bmatrix}^{-1}\begin{bmatrix}\Gamma_1 Z_{cp}\\ 0\end{bmatrix} \tag{7}$$

More information on the segmentation method can be found in [10].

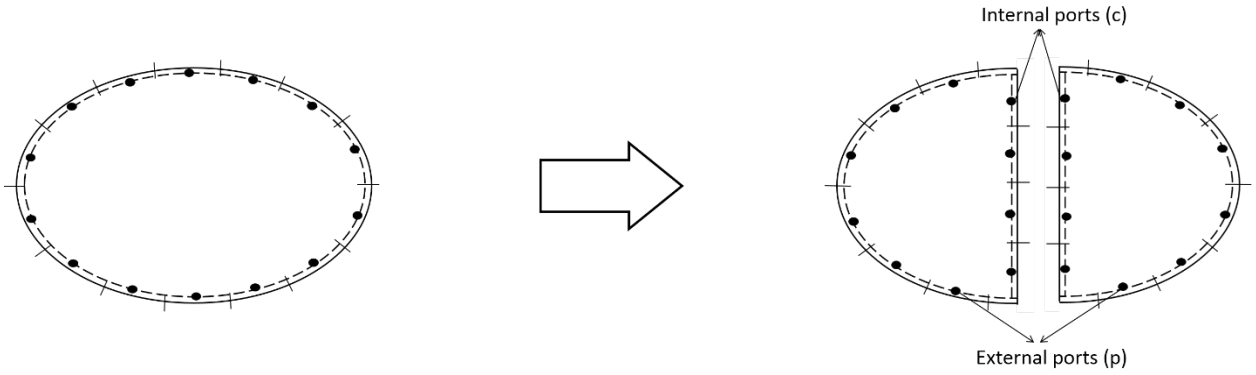


Fig. 13. Segmentation of planar configuration

3C. Preliminary Results

We consider the test case using a small Rotman lens-like structure as shown in Fig. 14 where 8 ports reside on a 43 mils thick Duroid 6002 $\left(\varepsilon_r = 2.94, \tan\delta = 0.0015\right)$ backed by a ground plane. The structure is decomposed into 9 segments: i.e. one lens region without taper and eight taper sections connecting to the lens. The *Z* matrices for nine segments are computed independently by using the contour integral method, and the overall *Z* matrix of the structure is obtained by using the segmentation method as described in section 3B. Then the *S*-parameters of the structure is calculated by transforming the overall *Z* matrix. The structure is analyzed at 41 equally spaced frequencies from 1 to 5GHz. The test results are compared with FEKO, a commercial EM software package. A comparison of return loss and port coupling magnitudes and phases are shown in Figs. 15-18. We observe that the results are slightly different between FEKO and the contour integral method. This is understandable since the contour integral method assumes the fields are constant in the *z* direction.

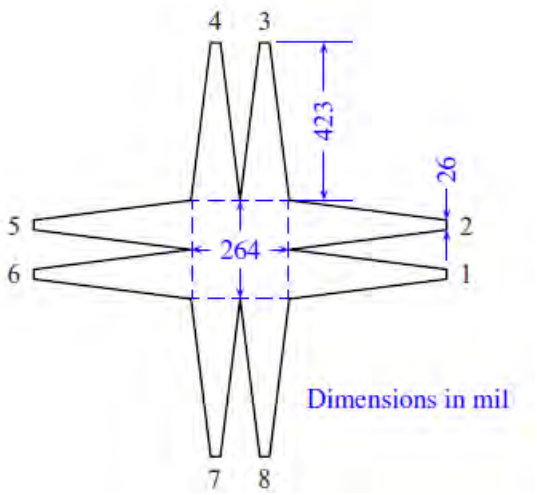


Fig. 14. Model used for validation

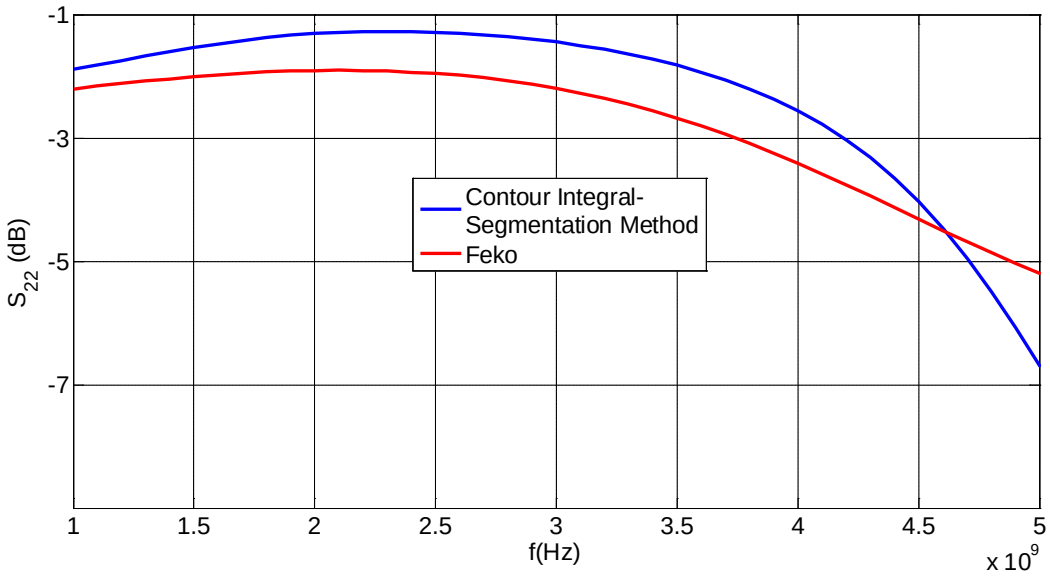


Fig. 15. Port 2 reflection magnitude

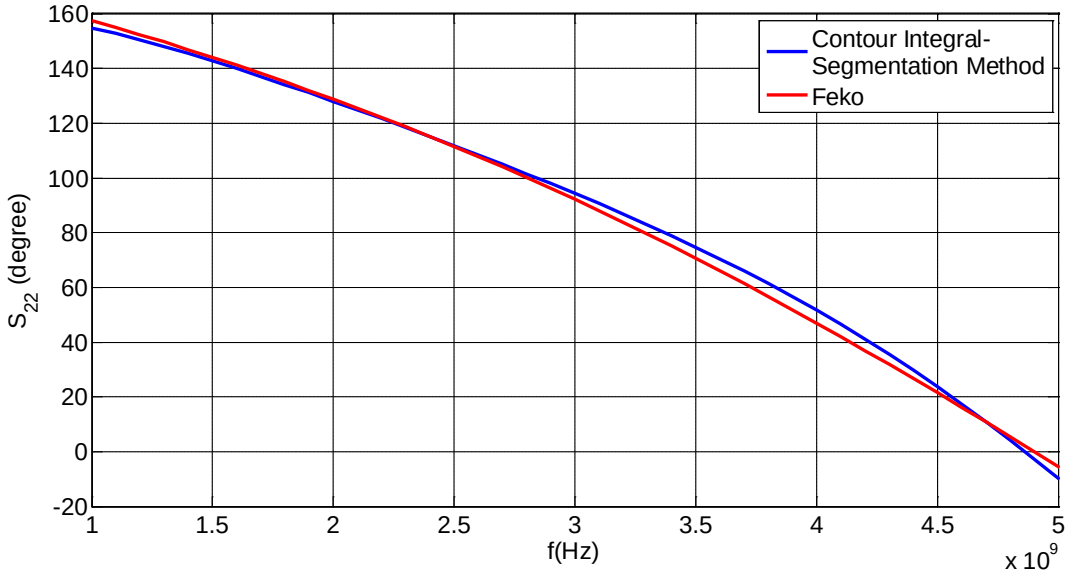


Fig. 16. Port 2 reflection phase

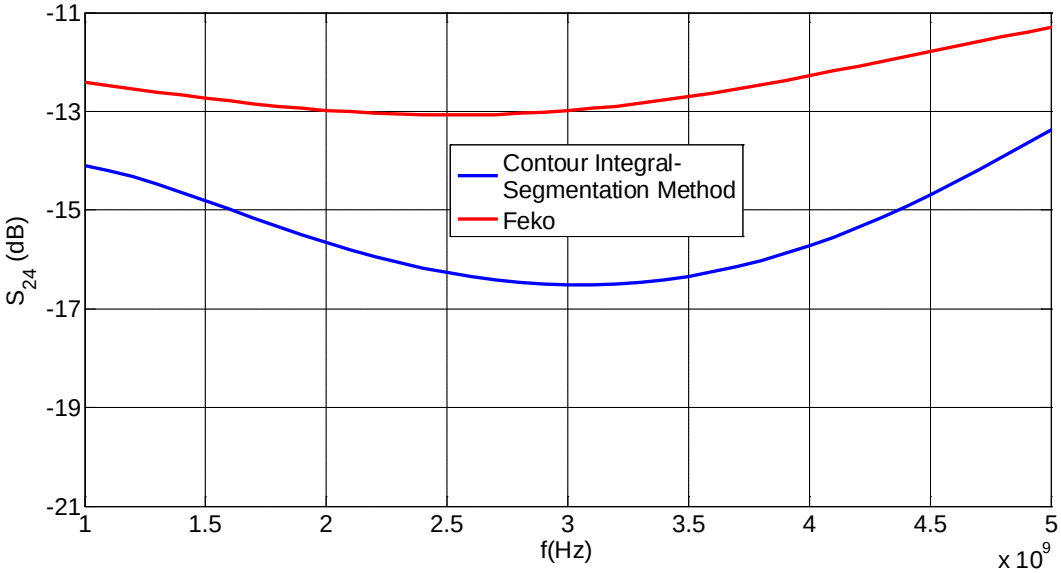


Fig. 17. Port 2-4 coupling magnitude

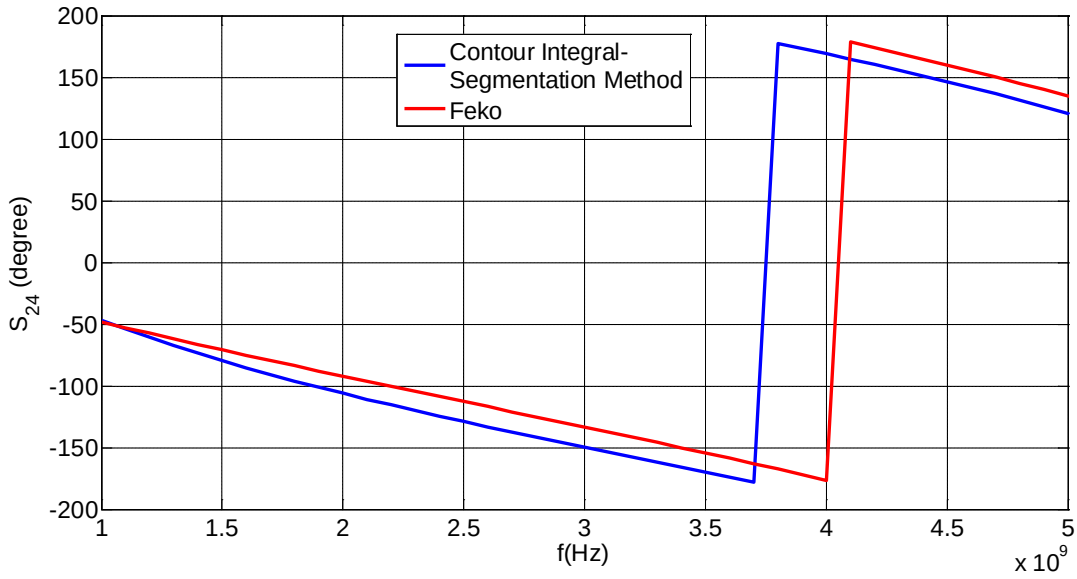
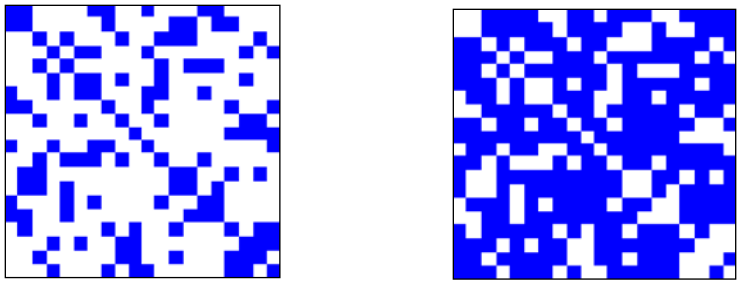


Fig. 18. Port 2-4 coupling phase.

Task 4: Parallelization of full-wave techniques on hybrid platforms

4A. Method of Moments (MoM) Enhanced with Fast Multi-pole Method (FMM)

The Fast Multipole Method (FMM), which was first introduced by Rokhlin, [11] as an augmentation to MoM, is one of the well-known techniques to reduce the computational complexity of electrically large problems without a significant loss of accuracy. In FMM, the edges in the mesh of a given structure are classified into local groups. For a mesh size of N edges, M localized groups are formed such that each group supports approximately N/M edges. The groups are categorized as near and far, based on their spatial proximity. FMM allows the system matrix to be split into two components, Z_{near} and Z_{far}, describing the near and far interactions among the edges, respectively. The interactions of edges within a group or within neighbor groups constitute the sparse Z_{near} matrix while the remaining interactions correspond to the Z_{far} matrix, as depicted in Fig. 19. The Z_{near} matrix is computed using conventional MoM and stored in memory. The components of Z_{far} are computed as needed through a fast matrix vector multiplication (MVM), which requires computations of radiation, T^E, receive, R^E, and translation functions, T_L. These functions are calculated over a set of K directions to carry out the unit sphere integration [11].



a) Z_{near} b) Z_{far}

Fig. 19. Sparse Z_{near}, and Z_{far} matrices.

4B. MoM-FMM Implementation on hybrid CPU-GPU platforms

The FMM basically consists of three main steps: pre-processing, processing, and post-processing. The pre-processing step involves importing the geometry mesh and clustering edges into groups using the regular grid-based group spacing. The algorithm computation depends only on non-empty groups. The processing step includes two phases, namely setup and solution of linear equation. The setup phase involves four calculation tasks, (i) Z^{near} matrix, (ii) radiation/receive functions T^{E}/R^{E}, (iii) translation matrix T_L, and (iv) V vector. Iterative methods, e.g. BiCGSTAB, are employed for solving the linear equation. The matrix-vector multiplications (MVMs), which dominate most of the computation in the solving phase, consists of calculating near and far MVMs. The far MVM comprises aggregation, translation, and disaggregation stages. Finally, the electromagnetic quantities of interest, e.g. scattered fields, are calculated in the post-processing step.

Based on our profiling results, we focus our implementation on the most computationally intensive step, i.e. the processing step. We notice that the translation matrix and the Z^{near} matrix are the critical factors that govern the memory requirement, and the translation stage and the near MVM are the two most time-consuming stages in the FMM. With a careful analysis of the entire algorithm, we thus develop a hybrid CPU-GPU implementation as depicted in Fig. 20. This strategy allows concurrently computing the translation and the near interactions while minimizing the memory usage by storing only necessary quantities, and calculating all others on-the-fly.

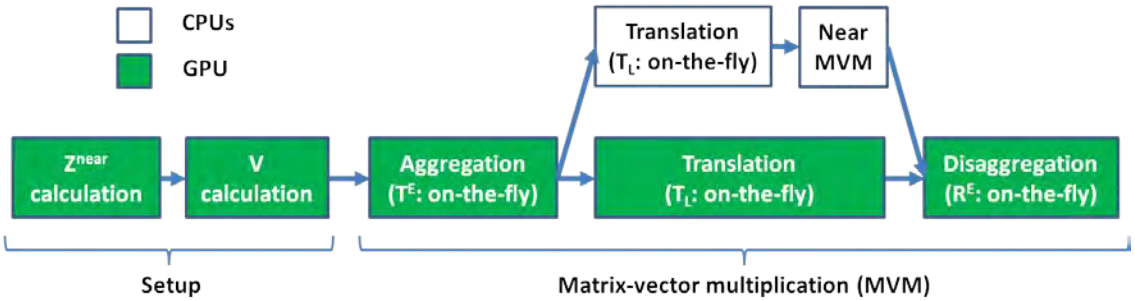


Fig. 20. Flowchart of FMM processing step using CPUs and GPU.

4B.1. Workload Distribution

Three parallelization levels are involved in the implementation: (i) between nodes using MPI, (ii) among CPU cores and GPU in each node using the POSIX pthreads model, and (iii) on GPU using the CUDA model. The balanced workload distributions among nodes are achieved by two multi-node parallelization schemes. The first scheme involves the equal distribution of *M* groups among *n* computing nodes. We define this technique of data distribution as group-based distribution. The group-based distribution is applied for the near interactions and the *V* vector. The second scheme involves the distribution of *K* independent direction computations among the nodes which is defined as direction-based distribution. This is applied for the far interactions, i.e. aggregation, translation and disaggregation, due to the fact that each direction is completely independent of each other.

Within each node, two parallelization schemes are developed for the hybrid CPU-GPU environment. For the stages only using GPU, the CUDA thread-block model is exploited to calculate the workloads assigned to that node. For the translation stage and near MVM, the pthreads is utilized on CPU cores in parallel with CUDA model. Each node is currently assigned approximately *K*/*n* translation matrices each of which has *M* rows. A combined group-direction distribution is employed in which the assigned M**K*/*n* rows of translation matrices are distributed among CPU cores and GPU. Given *P* as the ratio of a single CPU core computation time versus a single GPU time in performing the translation stage, we could distribute the M**K*/*n* rows of translation matrices (known as W_0^{transl}) among the processors such that the amount of time spent on CPUs (translation and near MVM) and GPU (translation) is balanced, i.e. $T_{GPU}^{transl} = T_{CPU}^{transl} + T_{CPU}^{ZnearI}$.

$$W_0^{transl} = W_{GPU}^{transl} + n_{CPU} W_{CPU}^{transl}, \tag{8}$$

$$T_{GPU}^{transl} = T_{CPU}^{transl} + T_{CPU}^{ZnearI} \Rightarrow \frac{W_{GPU}^{transl}}{B_{GPU}^{transl}} = \frac{W_{CPU}^{transl}}{B_{CPU}^{transl}} + T_{CPU}^{ZnearI}, \tag{9}$$

where W_{CPU}^{transl} and W_{GPU}^{transl} are the workload assigned to each CPU core and GPU , respectively, in the given node; B_{CPU}^{transl} and B_{GPU}^{transl} are the processing throughput of a CPU core and GPU , respectively; and n_{CPU} denotes the required number of CPU cores; T_{CPU}^{ZnearI} is the execution time to perform near MVM on CPUs.

As a result of equations (8) and (9), we could distribute the work among the processors proportionally to their relative speed/throughputs as follows:

$$W_{CPU}^{transl} = \frac{W_0^{transl} - \gamma}{P + n_{CPU}} \text{ , and } W_{GPU}^{transl} = \frac{W_0^{transl} P}{P + n_{CPU}} + \frac{\gamma n_{CPU}}{P + n_{CPU}}, \quad (10)$$

where $P = \frac{B_{GPU}^{transl}}{B_{CPU}^{transl}}$ and $\gamma = T_{CPU}^{ZnearI} B_{GPU}^{transl}$.

Since the translation stage is performed on GPU and CPUs, we can reduce the execution time as compared to the case of using only GPU. Based on the aforementioned workload distribution, the relative decrease time can be theoretically estimated as follows:

$$\frac{\Delta T}{T_{o,GPU}^{transl}}(\%) = \frac{T_{o,GPU}^{transl} - T_{GPU}^{transl}}{T_{o,GPU}^{transl}}(\%) = \left(\frac{n_{CPU}}{P + n_{CPU}} \right) \left(\frac{W_0^{transl} - \gamma}{W_o^{transl}} \right) \times 100, \quad (11)$$

where $T_{o,GPU}^{transl}$ is the time for GPU to complete the node-assigned workload W_0^{transl} .

4B.2. Hybrid Implementation

In this strategy, the Z^{near} matrix and the V vector are calculated once in the setup phase and stored in CPU memory. The readers are referred to our previous work in [12]-[14] for detailed CUDA implementations of Z^{near} calculation and V vector calculation. This section focuses on the implementation of matrix-vector multiplication on the GPU cluster, see Fig. 21. The main CPU thread is responsible for launching CUDA kernel for the aggregation stage on GPU using on-the-fly approach, and for setting up the pthreads environment for subsequent calculations on CPUs. The total number of parallel CPU threads per node is set to be equal to the number of CPU cores in a given node. The CPUs will then calculate one part of the translation (on-the-fly calculation) followed by the near MVM while the GPU concurrently performs the remaining of the translation on-the-fly. Finally, a CUDA kernel is invoked for the disaggregation stage on GPU using on-the-fly approach.

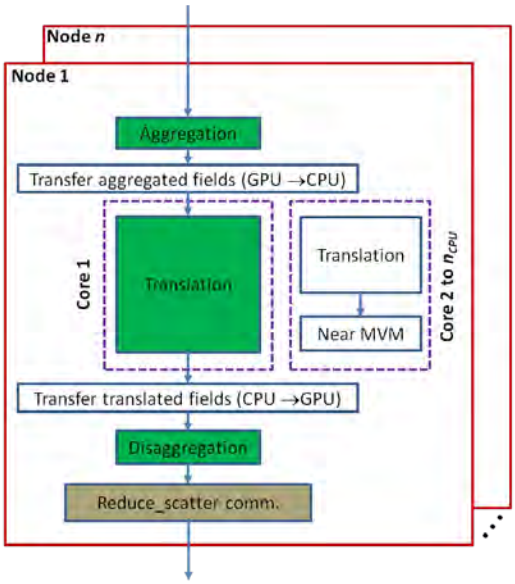


Fig. 21. Hybrid matrix-vector multiplication (MVM) using CPUs and GPU.

The matrix-vector multiplication inter-node communication at two steps: (i) before starting the MVM to update the estimated values for the unknowns among the nodes; (ii) after the disaggregation stage of the MVM to update the $Z_{far}I$ results among the nodes. It should be noted that, the presented hybrid CPU-GPU implementation requires extra data transfers between CPU and GPU memory spaces. More specifically, at each MVM, the aggregated fields of required directions are copied to the CPU before the translation occurs in CPU cores, and the resultant translated fields are transferred back to GPU for the disaggregation stage.

4B.3. Experimental Results

The GPU cluster platform used in our experimental work consists of 13 computing nodes, each of which is populated with dual Intel processors X5650 (6 cores per CPU) with 48 GB of memory. Each node is also equipped with an Nvidia Tesla M2090 GPU card with 6GB of GPU memory. The nodes are interconnected through Infiniband. The cluster populates the Native POSIX Thread Library (NPTL) 2.5, CUDA v4.2, and MVAPICH2 v1.8.1.

In our hybrid CPU-GPU implementation, the FMM is applied for the Electric-Field Integral Equation (EFIE) formulation and the bi-conjugate gradient stabilized method (BiCGSTAB) with diagonal preconditioning is used as the iterative solver. The single precision floating point representation is utilized for all experiments. Since the memory requirement mainly depends on the Z^{near} matrix which is $O(N)$ where N is the number of unknowns, the hybrid implementation can handle the problem size up to 10 million of unknowns.

We first validate the accuracy of the implementation by calculating the radar cross section (RCS) values for two cases: (i) a PEC sphere of diameter 31.5λ (discretized with ~2M unknowns), (ii) a

PEC sphere of diameter 70λ (discretized with ~10M unknowns). Each sphere is illuminated by an x-polarized normally incident field. The RCSs are compared with the results using Mie scattering and plotted in range of 140° to 180° as shown in Fig. 22 where 180° indicates the direction of forward scattering. For quantitative evaluation, a relative error is defined using the vector norm as follows

$$e(\%) = \frac{\left\| RCS_{FMM} - RCS_{Mie} \right\|_2}{\left\| RCS_{Mie} \right\|_2} \times 100. \tag{12}$$

It can be observed that the hybrid CPU-GPU results and the analytical Mie solutions show good agreements with a relative error of 1.7% and 5.9% for the RCSs of the 31.5λ-diameter sphere and RCS of the 70λ-diameter sphere, respectively. We note that a rather high relative error for the 70λ-diameter sphere case is due to the breakdown of the BiCGSTAB method after 428 iterations. Since the BiCGTAB method's convergence may be irregular, and there is a possibility that the method will break down [15], a remedy to this problem is to use a more robust iterative method, such as the Generalized Minimal Residual Method (GMRES), which will be investigated in our future work.

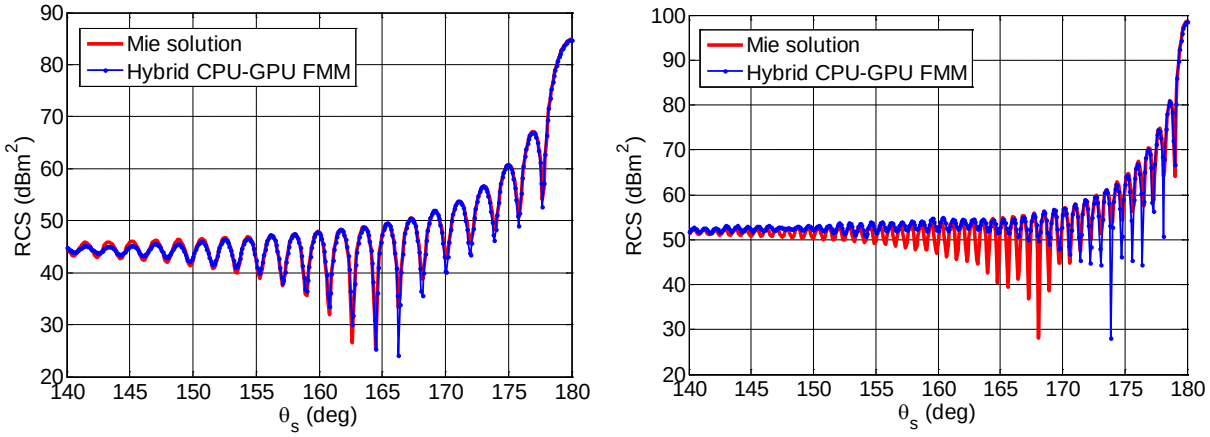


Fig. 22. (a) RCS of a 31.5λ-diameter PEC sphere, (b) RCS of a 70λ-diameter PEC sphere.

The performance of our hybrid CPU-GPU cluster implementation is evaluated in reference to an implementation on a single node with 12 CPU cores. Two metrics are used for the performance evaluation: (i) speedup, and (ii) scalability. The speedup is defined as the ratio of time required by multiple-node CPU-GPU implementation with respect to a single-node CPU implementation. Scalability is the normalized speedup of multiple nodes in reference to a single node. In our analysis, we consider the total execution time which is the sum of the computation time and the overhead associated with all communications between GPUs and CPUs. Two hybrid implementation cases are considered: (1) only GPU is used to perform the translation, and (2) CPUs and GPU jointly perform the translation (see Fig. 21).

The fixed-workload model (Amdahl's Law) is employed in our performance analysis. The sphere diameter is chosen as d=31.5λ (~2M unknowns) which fully utilizes the CPU and GPU memory on a single node. It is observed in Fig. 23 that the hybrid case (1) achieves a speedup factor of 20 on a single node. When the hybrid case (2) is used, a portion of the translation is assigned to compute on CPU cores in parallel with GPU. As the workload is distributed properly, the execution time spent on CPU cores and GPU is balanced which leads to a decrease in the total time (5% of decrease). Hence, the speedup increases from 20 to 21. However, it does not show a significant improvement because the ratio of GPU and CPU processing throughputs is approximately 200 which is much higher than the total number of 12 cores in the CPU system. As the number of nodes increases, each node processes less workload which results in a decrease of the total execution time. It can be seen from Fig. 23 that the speedup can be up to 256 and 262 for hybrid case (1) and (2), respectively, on 13 nodes.

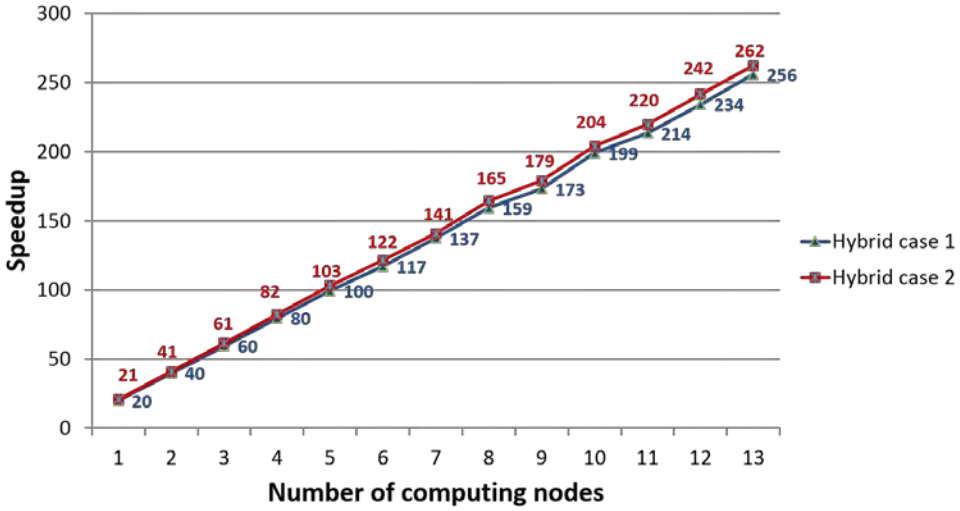


Fig. 23. Speedup of CPU-GPU implementations vs. single-node 12-core CPU implementation (CPU exec. time ≈ 4.1 hours, 20 iterations).

In order to investigate the scalability of the hybrid CPU-GPU implementation, we compare how the speedup improves with increasing computing nodes as shown in Fig. 24. It can be seen that both hybrid implementation cases scale closely to the theoretical linear expectation. This good scalability results from the fact that our implementation has efficiently parallelized the algorithm and reduced the inter-node communication overhead.

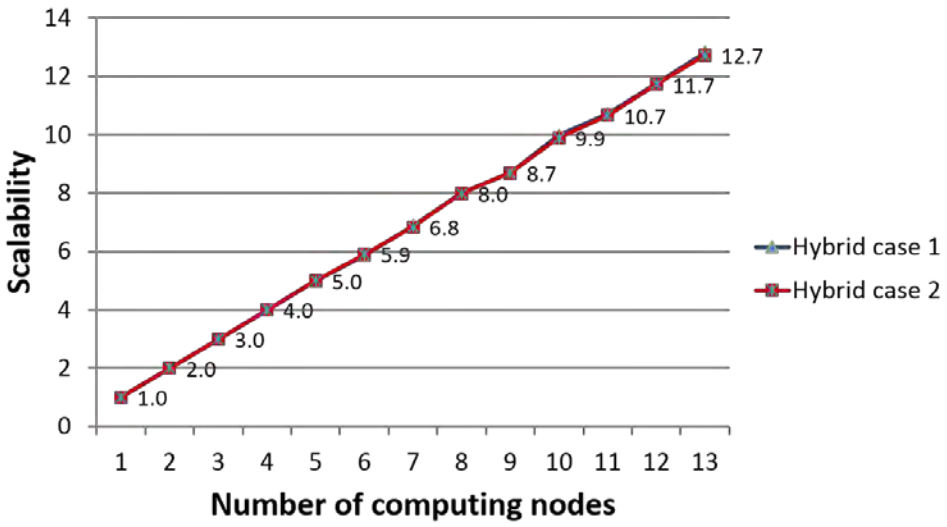


Fig. 24. Scalability of hybrid CPU-GPU implementations.

Conclusions:

Four different tasks have been investigated and completed successfully as part of this research effort. As a result of our work in this program our group is now able to solve for radiation and scattering problems that involve electrically large structures up to 10 Million unknowns using our 13 node GPU cluster.

We believe we can solve for larger problem sizes by implementing the MLFMA technique on a hybrid GPU-CPU platform. Furthermore, we have initiated numerical methods for iterative solutions of multi-domain problems that can support the MLFMA development to achieve solutions for larger problems. Our parallelizable optimization tools can also be utilized in the design and optimization of such structures.

References

[1] M. Dorigo, V. Maniezzo, and A. Colorni, “The Ant System: Optimization by a Colony of Cooperating Agents,” IEEE Trans. Systems, Man, and Cybernetics, Part B, Vol: 26 ,No. 1, pp. 1-13, 1996.

[2] R.C. Eberhart and Y. Shi, “Particle swarm optimization: Developments, applications, and resources,” Proc. 2001 Conf. Evolutionary Comp., pp. 81–86.

[3] Johnson, J. M. and Y. Rahmat-Sami, “Genetic Algorithm Optimization for Aerospace Electromagnetic Design and Analysis,” Proc. Aerospace Applications Conf., Vol. 1, 1996, pp. 87-102.

[4] Zhao, K., Rawat, V., and J. F. Lee, “A domain decomposition method for electromagnetic radiation and scattering analysis of multi-target problems,” *IEEE Trans. Antennas Propagat.*, vol. 56, no.8, pp. 2211-2221, 2008.

[5] Al Sharkawy, Mohamed H., Veysel Demir, and Atef Z. Elsherbeni. "The iterative multi-region algorithm using a hybrid finite difference frequency domain and method of moment techniques." *Progress In Electromagnetics Research.*, vol. 57, pp. 19–32, 2006.

[6] Q. Nguyen and O. Kilic, “Electromagnetic Scattering from Multiple Domains Using a Hybrid Numerical and Analytical Solution,” *ACES*, Jacksonville, FL, USA, 23-27 March 2014.

[7] Q. Nguyen and O. Kilic, “A Hybrid Method for Electromagnetic Scattering from Multiple Conducting Objects,” *APS/URSI*, Memphis, TN, USA, 6-12 July 2014.

[8] W. Rotman and R. Turner, "Wide-angle Microwave Lens for Line SourceApplications," *IEEE Transactions on Antennas and Propagation*, vol. 11, pp.623-632, 1963.

[9] D R. C. Hansen, "Design Trades for Rotman lenses," *IEEE Transactions on Antennas and Propagation*, vol. 39, pp. 464-472, 1991.

[10] T. Okoshi. *Planar circuits for microwaves and lightwaves.* Springer, 1985.

[11] R. Coifman, V. Rokhlin, S. Wandzura, “The fast multipole method for the wave equation: a pedestrian prescription”, IEEE Antennas Propagat. Mag., vol 35, no.3, pp. 7-12, June 1993.

[12] Q. Nguyen, V. Dang, O. Kilic, and E. El-Araby, “Parallelizing Fast Multipole Method for Large-Scale Electromagnetic Problems using GPU Clusters,” *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 868-871, July 2013.

[13] V. Dang, Q. Nguyen, and O. Kilic, “Fast Multipole Method for Large-Scale Electromagnetic Scattering Problems on GPU Cluster and FPGA-Accelerated Platforms,” *The Applied Computational Electromagnetics Society (ACES) Journal*, vol. 28, no. 12, pp, 1187-1198, Dec. 2013.

[14] V. Dang, Q. Nguyen, and O. Kilic, “GPU Cluster Implementation of FMM-FFT for Large-Scale Electromagnetic Problems,” *IEEE Antennas Wireless Propag. Lett.*, vol. 13, pp. 1259-1262, June 2014.

[15] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, 2nd ed., Philadelphia, PA: SIAM, 1994.